

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-031126

(43)Date of publication of application : 02.02.1999

(51)Int.Cl. G06F 15/00  
G06F 15/00  
G06F 13/00

(21)Application number : 10-122395

(71)Applicant : INTERNATL BUSINESS MACH CORP  
<IBM>

(22)Date of filing : 01.05.1998

(72)Inventor : DIAS DANIEL MANUEL  
IYENGAR ARUN KWANGIL

(30)Priority

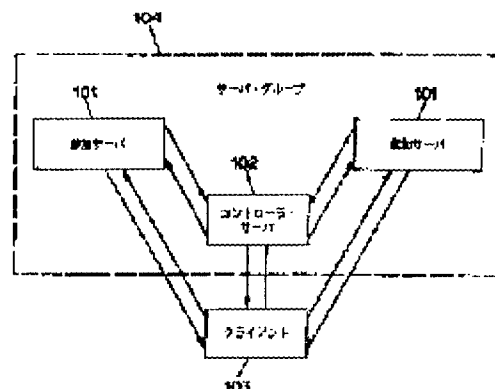
Priority number : 97 854010 Priority date : 08.05.1997 Priority country : US

## (54) METHOD FOR ADJUSTING OPERATION BETWEEN SERVER GROUPS

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a method that adjusts one time authentication of a client between sets of a store in order to execute transaction at an optional store or an optional subset of a store.

**SOLUTION:** In a client/server environment which includes participating servers 101 and a controller server 102 and has a server group 104 in which each server 101 can communicate with the server 102, this method with which operations are adjusted between servers of the group 104 includes (1) a step where communication is performed between a client 103 and a server of the group 104 by using a limited procedure or an undefined protocol through which state information is exchanged between the client 103 and the server, (2) a step where the client 103 transfers a request for an operation to one of servers and (3) a step where plural servers communicate with the server 102 in response to the request and the operations of at least some of servers are adjusted.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-31126

(43)公開日 平成11年(1999) 2月2日

(51)Int.Cl. <sup>9</sup>	識別記号	F I
G 0 6 F 15/00	3 1 0	G 0 6 F 15/00 3 1 0 A
	3 3 0	3 3 0 C
13/00	3 5 7	13/00 3 5 7 Z

審査請求 未請求 請求項の数30 O L (全 18 頁)

(21)出願番号 特願平10-122395

(22)出願日 平成10年(1998) 5月1日

(31)優先権主張番号 08/854010

(32)優先日 1997年5月8日

(33)優先権主張国 米国 (US)

(71)出願人 390009531

インターナショナル・ビジネス・マシー  
ズ・コーポレーションINTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATIONアメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72)発明者 ダニエル・マニュエル・ディアス

アメリカ合衆国10541、ニューヨーク州マ  
ホバック、ピケ・ブレース 16

(74)代理人 弁理士 坂口 博 (外1名)

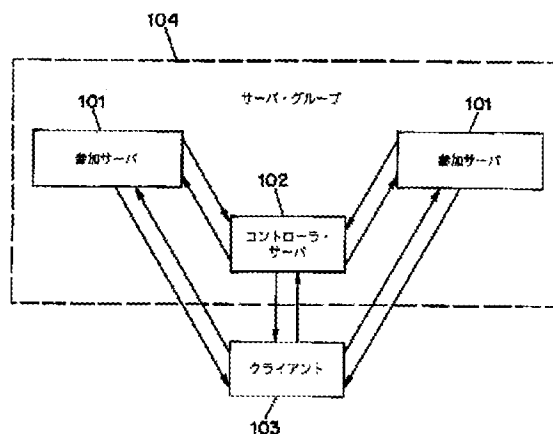
最終頁に続く

(54)【発明の名称】 サーバ・グループ間で操作を調整する方法

(57)【要約】

【課題】 任意のストアまたはストアの任意のサブセ  
ットにてトランザクションを実行できるように、ストアの  
セット間でクライアントの1回の認証を調整する方法を  
提供すること。

【解決手段】 参加サーバ及びコントローラ・サーバを  
含み、各参加サーバがコントローラ・サーバと通信でき  
るサーバ・グループを持つクライアント/サーバ環境  
で、サーバ・グループのサーバ間で操作を調整する方  
法は、1) クライアント及びサーバ・グループのサーバの  
間でクライアント及びサーバの間で状態情報を受け渡し  
手順が制限されたまたは未定義のプロトコルを使用して  
通信するステップ、2) クライアントからサーバの1つ  
に操作のリクエストを転送するステップ、及び3) リク  
エストに応答して複数のサーバ間でコントローラ・サー  
バと通信し、少なくともいくつかのサーバの操作を調整  
するステップを含む。



## 【特許請求の範囲】

【請求項1】1つ以上の参加サーバ及び1つ以上のコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント／サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡し手順が制限されたまたは未定義のプロトコルを使用して前記クライアント及び前記サーバの間で通信するステップと、  
前記クライアントから前記サーバの1つに操作のリクエストを転送するステップと、  
前記リクエストにตอบสนองして複数のサーバ及び少なくとも1つのコントローラ・サーバの間で通信して少なくともいくつかのサーバの操作を調整するステップと、  
を含む、方法。

【請求項2】参加サーバとコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント／サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、  
クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡し手順が制限されたまたは未定義のプロトコルを使用して前記クライアントと前記サーバとの間で通信するステップと、  
認証を要するクライアントによる第1操作のリクエストを転送するステップと、  
前記クライアントによりクライアント識別情報を与えるステップと、  
前記クライアント識別情報をコントローラ・サーバにより確認するステップと、  
コントローラ・サーバによりクライアント認証情報を作成するステップと、  
サーバにより前記クライアント認証情報を維持するステップと、  
前記クライアント認証情報を会話で保存するためにサーバにより手段を利用するステップと、  
認証を要する前記サーバによる第2操作のリクエストを前記サーバの1つに転送するステップと、  
前記会話で保存された前記クライアント認証情報とサーバによって維持されたクライアント認証情報を使用して前記クライアントを認証するステップと、  
を含む、方法。

【請求項3】1つ以上の参加サーバ及び1つ以上のコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント／サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、  
クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡し手順が制限されたまた

は未定義のプロトコルを使用して前記クライアント及び前記サーバの間で通信するステップと、  
参加を要する操作のリクエストを複数のサーバにより前記クライアントから前記サーバの1つに転送するステップと、  
前記リクエストにตอบสนองして前記サーバと少なくとも1つのコントローラ・サーバの間で通信して前記複数のサーバの操作を調整するステップと、  
を含む、方法。

10 【請求項4】クライアントによりクライアント識別情報を与える前記ステップは前記クライアントによる情報の手動入力により行われる、請求項2記載のサーバ間で操作を調整する方法。

【請求項5】クライアントによりクライアント識別情報を与える前記ステップは永続メモリに格納された情報を取得することにより行われる、請求項2記載のサーバ間で操作を調整する方法。

20 【請求項6】クライアント識別情報を確認する前記ステップは、1つ以上のコントローラ・サーバによって維持されたクライアント・アカウント情報を取得し、前記クライアント識別情報を前記クライアント・アカウント情報と比較するステップを含む、サーバ間で操作を調整する、請求項2記載の方法。

【請求項7】クライアント識別情報を確認する前記ステップは、前記サーバ・グループの外部の認証機関サーバと通信するステップを含む、サーバ間で操作を調整する、請求項2記載の方法。

30 【請求項8】1つ以上のサーバにより前記クライアント認証情報を維持するステップは、前記クライアント認証情報を前記サーバ・グループの2つ以上のサーバに複製するステップを含む、サーバ間で操作を調整する、請求項2記載の方法。

【請求項9】前記クライアント認証情報を会話で保存する手段は動的引数組み込みである、サーバ間で操作を調整する、請求項2記載の方法。

【請求項10】クライアント認証情報を維持する前記サーバは少なくとも1つのコントローラ・サーバを含む、サーバ間で操作を調整する、請求項2記載の方法。

40 【請求項11】前記クライアント認証情報はセッションIDを含む、サーバ間で操作を調整する、請求項2記載の方法。

【請求項12】前記クライアントにより、認証を要する指定数のリクエストを1つ以上のサーバに転送するステップと、  
前記指定数のリクエストの間に1つ以上のコントローラ・サーバにより新しいセッションIDを含む新しいクライアント認証情報を作成するステップと、  
1つ以上のサーバにより前記新しいクライアント認証情報を会話で保存する手段を利用するステップと、  
50 を含む、サーバ間で操作を調整する、請求項11記載の

方法。

【請求項13】クライアント認証情報を保存する手段は、前記セッションIDを暗号化する手段を含み、前記会話で保存されたクライアント認証情報を使用することは暗号化されたセッションIDのデコードを伴う、サーバ間で操作を調整する、請求項1記載の方法。

【請求項14】前記セッションIDを暗号化する手段は、サーバとリクエストの特定の組み合わせについての有効な暗号化されたセッションIDを作成する、サーバ間で操作を調整する、請求項13記載の方法。

【請求項15】少なくとも1つのサーバにより前記クライアント識別情報を与えた前記クライアントのアドレスを維持するステップと、前記会話中に認証を要し前記クライアントにより要求された少なくともいくつかの操作にตอบสนองして、前記アドレスを使用して前記クライアントを認証するステップと、を含む、サーバ間で操作を調整する、請求項2記載の方法。

【請求項16】少なくとも1つのコントローラ・サーバは複数のノードに分散される、サーバ間で操作を調整する、請求項2記載の方法。

【請求項17】前記クライアント認証情報はセッションIDを含み、1つ以上サーバにより前記クライアント認証情報を維持するステップは、前記セッションIDを前記分散されたコントローラ・サーバの1つの全てのノードよりも少ないノードに格納し、前記セッションIDが格納された前記ノードの少なくとも1つのIDを前記会話で状態変数に保存するステップを含む、サーバ間で操作を調整する、請求項16記載の方法。

【請求項18】前記クライアント認証情報は前記分散されたコントローラ・サーバの1つの複数のノードに複製される、サーバ間で操作を調整する請求項16記載の方法。

【請求項19】少なくとも1つのコントローラ・サーバが複数のノードに分散され、前記クライアント・アカウント情報は前記分散されたコントローラ・サーバの1つの複数のノードに複製される、サーバ間で操作を調整する、請求項6記載の方法。

【請求項20】前記複数のサーバの操作を調整するステップは、前記クライアントによるリクエストにตอบสนองして少なくとも1つのコントローラ・サーバによりサーバのデフォルト・グループを定義するステップを含む、サーバ間で操作を調整する、請求項3記載の方法。

【請求項21】前記サーバのデフォルト・グループは、前記クライアントが会話中に通信した前記サーバ・グループのサーバを含む、サーバ間で操作を調整する、請求項20記載の方法。

【請求項22】前記サーバのデフォルト・グループは、サーバの重要サブセットに関して決定され、サーバの各重要サブセットは、トランザクション・タイプ、認証タ

イプ及び重要イベントに関して決定される、サーバ間で操作を調整する、請求項21記載の方法。

【請求項23】前記サーバのデフォルト・グループは、前記最後のグループ・トランザクションに関係したサーバに関して決定される、サーバ間で操作を調整する請求項20記載の方法。

【請求項24】前記操作のリクエストは複数のサーバから情報を集める操作を伴う、サーバ間で操作を調整する、請求項3記載の方法。

10 【請求項25】複数のサーバの参加を要する前記操作のリクエストは、有限な $n > 0$ のとき複数のサーバからの $n$ フェーズ・コミットを伴う、サーバ間で操作を調整する、請求項3記載の方法。

【請求項26】前記サーバ間で少なくとも1つのコントローラ・サーバと通信して前記複数のサーバの操作を調整するステップは、少なくとも1つのコントローラ・サーバ及び前記複数のサーバのうち少なくともいくつかのサーバとの間で通信するステップと、前記複数のサーバのうち少なくともいくつかのサーバからの応答を少なくとも1つのコントローラ・サーバによりマージするステップと、を含む、サーバ間で操作を調整する、請求項3記載の方法。

20 【請求項27】少なくとも1つの参加サーバは複数のコントローラと通信する、サーバ間で操作を調整する、請求項1乃至請求項3のいずれかに記載の方法。

【請求項28】コントローラ・サーバの階層が存在し、各コントローラ・サーバが下の階層レベルの参加サーバ及びコントローラ・サーバの間で操作を調整する、サーバ間で操作を調整する、請求項1乃至請求項3のいずれかに記載の方法。

【請求項29】前記クライアントにより少なくとも1つのサーバからプログラムを受信し実行することによって、前記クライアントが前記サーバ・グループの機能の少なくともいくつかを実行するステップを含む、サーバ間で操作を調整する請求項1乃至請求項3のいずれかに記載の方法。

40 【請求項30】前記参加サーバは互いに通信できない、サーバ間で操作を調整する請求項1乃至請求項3のいずれかに記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的にはサーバ・グループ間で操作を調整する方法に関し、特にサーバ・グループ間でクライアントの認証、情報の収集、グループ・トランザクション、及び複数フェーズ・コミットを調整する方法に関する。本発明の好適な実施例はWWW (World Wide Web) 上のトランザクションに関する。

【0002】

【従来の技術】電子取引は急激に増加しており、この傾向は特にWWWで顕著である。Webサイトに出品する企業も増えている。消費者はWebサイトで製品のオンライン・ショッピングができる。こうした独立した仮想ストアを延長する形で、仮想モールまでWebに登場し始めている。

【0003】現実のモールと同様、仮想モールも企業及び消費者の両方に資する機会を提供する。企業は、モール上に存在することから利益を得る可能性がある。例えば仮想モール上の仮想ストアは、初めは他のストアから商品を購入するためにアクセスした消費者の注意を引くこともできる。消費者は、モール上で消費及びサービスが統合され調整される機会から利益を得ることがある。仮想モールは、例えば多くの仮想ストアの商品及びサービスの総合ディレクトリを提供できる。また仮想モールは、モール上の全ての仮想ストアのための単独認証点（single point of authentication）となることもできる。更に仮想モールは、複数のストアが関係するグループ・トランザクションを調整することもできよう。

【0004】これまで、Webに現れた仮想モールのアプローチは2種類ある。第1に、独立した仮想ストアとのハイパーリンクを持つWebページを提供するモールがある。第2に、同じWebサイト内で複数の仮想ストアを提供しているモールがある。これら後者のモールは、IBMのワールド・アベニュー・システム（<http://mer.shop.ibm.com>）、及びNetscapeのマーチャント・システム（<http://home.netscape.com/comprod/products/iapps/capps/mersys.html>）などのシステムを利用している。

【0005】こうしたアプローチそれぞれに欠点がある。最初のアプローチは、URLの収集を除いて独立したストア間で調整を行わない。もう1つのアプローチは、独立したストア間の調整を行うが、仮想ストアは1つのWebサイト内で実現し管理しなければならないという制限がある。従って、第2のアプローチで仮想モールに加わるためには、Webに仮想ストアを確立した企業は、その実現形態を大きく変更する必要があるかもしれない。

【0006】よって、複数のWebサイトに分散できる独立した複数の仮想ストアを調整する方法が必要である。

【0007】

【発明が解決しようとする課題】本発明はサーバのグループ間で操作を調整する方法を対象とする。本発明の好適な実施例は、Web上の複数の仮想ストアの調整に関係するが、本発明は他の形態のネットワーク通信にも適用される。

【0008】本発明の目的は、クライアントが認証情報を求められた場合、認証を更に求められることなく、任意のストアまたはストアの任意のサブセットにてトラン

ザクションを実行できるように、ストアのセット間でクライアントの1回の認証を調整する方法を提供する。

【0009】本発明の他の目的は、複数のストアからの情報の統合を調整することである。例えばクライアントがアクセスした各ストアのショッピング・バスケットに含まれるアイテムを統合し、この情報の1つの表示をクライアントに与えることが望ましい。他の例として、クライアントが検索のためのキーワードを指定し、ストアの任意のサブセットにサーチ・リクエストを送り、組み合わせられたサーチ結果をクライアントに送ることができるようにすることが望ましい。

【0010】本発明の他の目的は、複数のストアが関係するグループ・トランザクションのリクエストを調整することである。例えばクライアントはグループ購入を要求でき、これにはストア全体でアイテムのセットについて"オール・オア・ナッシング（all-or-nothing）"の購入が関係する。このような購入では、例えば別々のストアから組み合わせのいいジャケット及びスラックスを選ぶといったことが関係することもある。この種のリクエストは、ストア全体で1回のコミットまたは複数フェーズ・コミットと共に行うことができる。

【0011】

【課題を解決するための手段】本発明のアーキテクチャはクライアント/サーバ環境であり、1つ以上の参加サーバ及び1つ以上のコントローラ・サーバを含むサーバ・グループがあり、参加サーバはそれぞれ少なくとも1つのコントローラ・サーバと通信できる。この環境で、本発明はサーバ・グループ内のサーバ間の操作を調整する方法を対象とする。方法のステップは、1）クライアント及びサーバの間で状態情報を受け渡す手順が制限された、または未定義のプロトコルを使用してクライアント及びサーバ・グループの少なくとも1つのサーバの間で通信するステップ、2）操作のリクエストをクライアントからサーバの1つへ転送するステップ、及び3）リクエストに応答して複数のサーバ間で少なくとも1つのコントローラ・サーバと通信し、少なくともいくつかのサーバの操作を調整するステップを含む。

【0012】本発明の好適な実施例は、会話中のクライアント認証の調整及びクライアントによって要求されたグループ・トランザクションの調整を含む。会話中のクライアント認証を調整するために、本発明に従った好適な方法は、1）クライアント及びサーバの間で状態情報を受け渡すために手順が制限されたまたは未定義のプロトコルを使用してクライアント及びサーバ・グループの少なくとも1つのサーバの間で通信するステップ、2）認証を必要とする第1操作のリクエストをクライアントからサーバへ転送するステップ、3）クライアントによりクライアント識別情報を与えるステップ、4）コントローラ・サーバによりクライアント識別情報を確認するステップ、5）コントローラ・サーバによりクライアン

ト認証情報を生成するステップ、6) サーバによりクライアント認証情報を維持するステップ、7) クライアント認証情報を会話で保存するためサーバにより手段を利用するステップ、8) 認証を必要とするクライアントによる第2操作のリクエストをサーバの1つに転送するステップ、及び9) 会話で保存されたクライアント認証情報及びサーバによって維持されるクライアント認証情報を使用してクライアントを認証するステップを含む。

【0013】クライアントが要求したグループ・トランザクションを調整するために、本発明に従った好適な方法は、1) クライアント及びサーバの間で状態情報を受け渡すため、手順が制限されたまたは未定義のプロトコルを使用して、クライアント及びサーバ・グループの少なくとも1つのサーバの間で通信するステップ、2) 複数のサーバによる参加を要求する操作のリクエストをクライアントからサーバの1つに転送するステップ、及び3) リクエストに回答してサーバ間で少なくとも1つのコントローラ・サーバと通信し、複数のサーバの操作を調整するステップを含む。

【0014】

【発明の実施の形態】本明細書及び特許請求の範囲で用いているとおり、“クライアント”という用語は、コマンドをコマンドに関連したタスクを実行する他のプログラムまたはアプリケーション（サーバと呼ぶ）に発行するプログラムまたはアプリケーションである。

【0015】本明細書及び特許請求の範囲で用いているとおり、“CGI（共通ゲートウェイ・インタフェース）”という用語は、WWWクライアントがWebサイトでのプログラムの実行を要求できるメカニズムである。WebサイトはCGIプログラムを実行し、プログラムの出力をクライアントに送る。

【0016】本明細書及び特許請求の範囲で用いているとおり、“接続法（continuation）”という用語は、クライアントがサーバに送ることのできる新しいリクエストであり、クライアントによる前のリクエストに回答してサーバにより与えられたものである。クライアント/サーバ通信での接続法の例としてハイパーテキスト・リンク（またはハイパーリンク）がある。クライアントがサーバにリクエストを出すときは、サーバはその応答に接続法を加えることができる。サーバにより与えられる接続法は任意の有効なリクエストでよいが、有益な接続法は、一般的にはクライアントによる最初のリクエストと論理的に関係する。

【0017】本明細書及び特許請求の範囲で用いているとおり、“会話”という用語は、クライアント及びサーバの間の通信のシーケンスであり、サーバは会話の中でクライアントからの各リクエストに対して1組の接続法で応答し、クライアントは常にサーバから与えられた1組の接続法から次のリクエストを拾う。WWWではハイパーテキスト・リンクは接続法の例であり、クライアント

は、サーバにより与えられたハイパーテキスト・リンクを辿るときにはサーバとの会話に関係する。具体的には、HTMLの一連のWebページ $h_1, h_2, \dots, h_n$ が会話をなすのは、1) ページ $h_1, h_2, \dots, h_n$ が全てクライアントによって表示された場合、及び2)  $1 < i \leq n$ のときの全ての $i$ について、ページ $h_i$ がページ $h_{i-1}$ 上のハイパーテキスト・リンクを辿ることによって取得された場合である。割り込みのない会話では、クライアントはただ $n-1$ のハイパーテキスト・リンクを辿り、ページ $h_1$ から $h_n$ を得るだけであり、同じ経路で戻る（バックトラックする）ことはない。割り込みのある会話では、クライアントは少なくとも1回はバックトラックする。バックトラックはクライアントが、1)  $1 < i \leq n$ のとき最初にページ $h_i$ にアクセスする、2) ハイパーテキスト・リンクを辿り、URLを明示的に要求することによってか、またはWebブラウザのキャッシュを使用することによって他のページを表示する、及び3) メモリから（ブラウザのキャッシュ等から） $h_i$ を再ロードすることによってページ $h_i$ に戻るプロセスをいう。

【0018】本明細書及び特許請求の範囲で用いているとおり、“ホームページ”という用語は、Webサイトに格納された情報までWWWユーザを案内する目次である。ホームページはしばしばマルチメディア・コンテンツを含む。

【0019】本明細書及び特許請求の範囲で用いているとおり、“ハイパーリンク”または“ハイパーテキスト・リンク”は、語句、アイコン、ピクチャ等に組み込まれ、選択すると起動されるURL（Universal Resource Locator）等のネットワーク・アドレスである。クライアントがハイパーリンクを起動すると、ハイパーリンクに対応した情報がクライアントに返され、Webブラウザを使用して表示される。

【0020】本明細書及び特許請求の範囲で用いているとおり、“HTML（ハイパーテキスト・マークアップ言語）”は、Webクライアントによって表示されるハイパーテキスト文書を作成し接続するためにWebサーバによって用いられる言語である。ハイパーテキスト文書の他の用法については、Bernsteinらによる1993年4月20日付、米国特許番号第5204947号、Bernsteinらによる1994年3月22日付、米国特許番号第5297249号、及びLewisによる1994年10月11日付、米国特許番号第5355472号を参照されたい。

【0021】本明細書及び特許請求の範囲で用いているとおり、“HTTP（ハイパーテキスト転送プロトコル）”は、コンピュータ間でデータを転送するためにWWWによって用いられるプロトコルである。HTTPによりクライアントはファイルを要求するか、またはCGIプログラムを起動することによって、サーバからデー

タを取得することができる。HTTPは無状態プロトコル、つまりクライアントからサーバへのリクエストは全て個別に扱われるプロトコルである。サーバはクライアントとの前の接続について記録を持たない。URLの先頭の"http:"は、使用されるプロトコルがHTTPであることを示す。

【0022】本明細書及び特許請求の範囲で用いているとおり、「インターネット」という用語は、TCP/IPプロトコルを使用して互いに接続されたコンピュータのネットワーク及びゲートウェイの集まりである。

【0023】本明細書及び特許請求の範囲で用いているとおり、「サーバ」という用語は、他のプログラムまたはアプリケーション（クライアントという）のコマンドでタスクを実行するプログラムまたはアプリケーションである。本明細書及び特許請求の範囲で用いているとおり、Webサイトはサーバにもなる。「サーバ」という用語はWebサイトをサポートするWebサーバと同義語ではない。

【0024】本明細書及び特許請求の範囲で用いているとおり、「トランザクション」という用語は、サーバに対してクライアントによって出されるリクエストを指し示す。

【0025】本明細書及び特許請求の範囲で用いているとおり、「TCP/IP（転送制御プロトコル/インターネット・プロトコル）」は、データをパケットに分け、パケットをネットワークでルーティングし、宛先でパケットを再構成するプロトコルの集まりである。データは電子メール・テキストからビデオまで任意の形でよい。

【0026】本明細書及び特許請求の範囲で用いているとおり、「URL（Universal Resource Locator）」は、Web上の情報を一意に識別するまたはアドレスする仕方である。URLは、Webでは電子メール・アドレスと同等とみなすことができる。URLはしかし、他の文書内の深いところに埋められている文書に属する場合は厄介にもなるが、しばしばハイパーリンクでアクセスすることができる。URLの一例として「http://www.arun.com:80/table.html」がある。

【0027】URLには4つのコンポーネントがある。左端のコンポーネントは使用されるプロトコルを指定する。これはURLの残りの部分とはコロンので分けられる。次のコンポーネントはターゲット・ホストのホスト名またはIP（インターネット・プロトコル）アドレスである。このコンポーネントは左側が"/"で区切られ、右側は"/"またはコロンので区切られる。3つ目のコンポーネントはオプションのポート番号である。このコンポーネントが用いられるときは、左側はコロンので、右側は"/"で区切られる。4つ目のコンポーネントは実際のファイル名またはプログラム名である。

【0028】上の例で、プロトコルは"http"、ホスト名

は「www.arun.com」、ポート番号は「80」、ファイル名は「table.html」である。この例の"http"はハイパーテキスト転送プロトコルを指し示し、ファイル名の拡張子「html」はハイパーテキスト・マークアップ言語を指し示す。

【0029】本明細書及び特許請求の範囲で用いているとおり、「Webブラウザ」は、WWWでハイパーテキスト文書を表示することのできるクライアント・プログラムである。Webブラウザはインターネットのツアー・ガイドとして働く。つまりビジュアルなデスクトップ、ディレクトリ、及びサーチ・ツールを備え、これらを使用してインターネットをナビゲートまたは「サーフ」することができる。

【0030】本明細書及び特許請求の範囲で用いているとおり、「Web」または「WWW（World Wide Web）」は、インターネット上のハイパーテキスト文書の集まりである。ハイパーテキスト文書は、他の文書へのハイパーリンクを含み、これを使用して文書から文書へナビゲートすることができる。実際には、インターネットで情報を探す際にWebを使用してサーバからサーバへ、またデータベースからデータベースへの切り替えを行うためにハイパーリンクを選択するが、ハイパーリンクは普通は文書内の強調表示された語句として表示される。Web上のリソースはURL（Universal Resource Locator）を通してアクセスできる。WebはHTML（ハイパーテキスト・マークアップ言語）を利用して、URLに対応した情報を表示し、他のURLへのポイント/クリック・インタフェースを与える。

【0031】図1に示すように、本発明の好適な実施例に従って用いられるアーキテクチャ全体は、クライアント103及びサーバ・グループ104で構成され、サーバ・グループ104は参加サーバ101及びコントローラ・サーバ102を含む。（コントローラ・サーバ102は本明細書では簡単にコントローラと呼ぶ。）参加サーバ及びコントローラの物理的位置は重要ではない。各参加サーバ101及びコントローラ・サーバ102は、ネットワークで接続された個別コンピュータ上にあってよく、また複数のサーバが1つのコンピュータにあってよい。更にコントローラや1つ以上の参加サーバはいくつかのコンピュータ上に同時に存在することも可能である。例えばこれは、コントローラや1つ以上の参加サーバがスケーラブルなWebサーバとして実現された場合である。（スケーラブルWebサーバについては、D. Diasらによる「A Scalable and Highly Available Web Server」、Proceedings of COMPCON '96を参照されたい。）

【0032】コントローラ・サーバ102はクライアント・アカウント情報（例えばユーザID、ユーザ名、パスワード、アドレス、電話番号）を維持する。図1に示すように、各参加サーバ101はコントローラ・サーバ

102と通信することができ、クライアント103は各参加サーバ101及びコントローラ・サーバ102と通信することができる。コントローラ及び参加サーバはHTTP等の標準プロトコルを使用して互いに通信できる。また独自プロトコルを使用して通信することもできる。

【0033】クライアント103は、参加サーバ101及びコントローラ・サーバ102と、クライアント状態情報を維持するための手順が制限されたまたは未定義のHTTP等の通信プロトコルを介して通信する。クライアント状態情報を維持する限定されたまたは未定義の手順とは、このプロトコルを使用したクライアント及びサーバの間の通信タイプの空でないサブセットについて、プロトコルによって定義された手順を使用した会話の中で、会話の残りの部分で状態変数をサーバにより後で取得できるように、クライアントのID、クライアント・パスワード等の状態変数を保存することが難しいまたは不可能なことである。

【0034】参加サーバが実行できるトランザクションは2種類ある。第1に参加サーバは単純トランザクションを実行できる。参加サーバは他の参加サーバ及びコントローラから独立してクライアントを代表して単純トランザクションを実行できる。よってクライアントは単純トランザクションを参加サーバから直接的に要求できる。第2に参加サーバはグループ・トランザクションを実行できる。グループ・トランザクションは、コントローラによる調整の下で1つ以上の参加サーバによって実行される。好適にはクライアントはコントローラからのみグループ・トランザクションを要求できる。

【0035】図2は、コントローラ・サーバ102が参加サーバ間でクライアント103の認証を調整する本発明の好適な実施例を示す。認証は、クライアント103がサーバ・グループ104のサーバに対して自体を識別するプロセスである。この好適な実施例でクライアント103は、会話中に会話の残りの部分でサーバ・グループの任意のサーバでトランザクションを実行するために、参加サーバまたはコントローラにより認証情報を一度要求されるだけである。

【0036】ステップ201でコントローラは、好適には参加サーバとオープン接続を確立する。またコントローラは参加サーバと通信する必要があるとき、その都度新しい接続を確立することもできる。しかしステップ201に示すようにオープン接続を維持すると、その都度新しい接続を確立する場合よりも性能が向上する可能性が大きい。

【0037】ステップ202でクライアント103は自体を認証せずにサーバ・グループと対話する。例えばクライアントはこの場合カタログをブラウズする。

【0038】ステップ204でクライアント103は、認証を必要とする会話の最初のトランザクションを実行

しようとする。

【0039】ステップ205でクライアントはユーザID401及びパスワードをHTML形式を介して入力する。機密を守るためにクライアント及びサーバ・グループの間で受け渡される情報は、好適にはSSL (Secure Sockets Layer) 等のシステムを使用して暗号化される。(SSLについては、A. O. Freier, P. Karlton, P. C. Kocherによる"The SSL Protocol Version 3.0", Internet Draft, March1996, <http://home.netscape.com/eng/ssl3/ssl-toc.html>を参照されたい)。

【0040】ステップ207でクライアントからコンタクトされたサーバは、クライアントが有効なユーザID401及びパスワードを入力したか確認する。クライアントからコンタクトされたサーバが参加サーバなら、参加サーバはコントローラにコンタクトしてユーザID401及びパスワードとを確認しなければならない。クライアントが有効なユーザID401及びパスワードを入力していない場合は、ステップ206でエラー処理が行われる。

【0041】ステップ206はいくつかの形で実現できる。一例としてクライアントにユーザID401及びパスワードを入力する新しい機会を与えることができる。他の例としては、クライアントに新しいユーザとして登録する機会を与えることもできる。その場合にクライアントは、名前、住所、電話番号等の他の情報も入力するよう促される。他の例として、会話をエラー・メッセージで中断することもできる。この最後の例は、エラー処理を扱うには好適な方法である。もちろん他の実現形態も可能である。

【0042】有効なユーザID401及びパスワードがステップ207で入力された場合、コントローラはステップ208でセッションID402を生成する。セッションIDは、会話中にクライアントを認証するために用いられる。セッションIDは、悪意あるユーザがセッションIDを類推しにくいように、十分に大きいキー・スペースからランダムに選択される。コントローラはユーザID及びセッションIDをデータベースに格納する。(ファイル・システム等他の記憶方法も使用できる)。現在のセッションIDは、特定のユーザIDに対して前に格納されたセッションIDを無効にする。

【0043】ステップ208で、クライアントからコンタクトされたサーバは、会話でユーザID401及びセッションID402を状態変数として保存する。本明細書及び特許請求の範囲で用いているとおり、会話で状態情報を保存するとは、サーバ・グループの任意のサーバが後で会話中にクライアントと通信するときに情報にアクセスできるように情報を保存することをいう。

【0044】ステップ208でサーバはまたそのIDをアクセス・ノード状態変数403にエンコードし、会話でアクセス・ノード状態変数を保存する。アクセス・ノ



ード状態変数403は、認証の間にクライアントによりアクセスされた参加サーバを表す1つ以上の状態変数で構成される。1つの変数を使用して、アクセスした全てのサーバを表すことが可能である。しかし複数の変数を使用してより詳細な情報を表すことが望ましい場合もある。例えば認証の前にアクセスした参加サーバ、認証の後にアクセスした参加サーバ、単純トランザクションが実行された参加サーバ等である。

【0045】クライアントを認証するためにコントローラがそのデータベースに格納する状態情報と、会話で保存される状態情報は必ずしも同一ではない。他より若干多いかまたは少ない状態変数が格納されることもある。しかしそれでも、それぞれに格納される少なくともいくつかの変数は同じである。例えば好適な実施例で、ユーザID及びセッションIDの変数は両方ともコントローラのデータベースに格納され、会話で保存される。本明細書及び特許請求の範囲で用いているとおり、“クライアント認証情報”という用語は、サーバによって格納され、会話でも保存される1組の変数をいう。

【0046】会話で状態変数を保存する好適な方法は、動的引数組み込みである。動的引数組み込みでは、サーバがクライアントに送る接続法に状態情報を再帰的に組み込む。この方法によると、クライアントが接続法を選択するときには、クライアントは状態情報を接続法の一部としてサーバに送り返す。そこでサーバは必要に応じて状態情報を調べ、リクエストを処理し、クライアントに送られる接続法の次のセットに状態情報をもう一度組み込む。

【0047】動的引数組み込みは、WWWのコンテキストでは、ハイパーリンクに変更を加えて状態情報を保存する。一般的にハイパーリンクは変更されて、引数エンベッタと呼ばれる特別なプログラムが起動される。引数エンベッタは、ハイパーリンクの保存された状態情報をWebサイトの他のCGIスクリプトに受け渡す。更に引数エンベッタは、得られたCGIスクリプトの出力の全てのハイパーリンクを変更して自体を起動する。従って、動的引数組み込みにより状態が保存された会話を辿るクライアントには、全てが引数エンベッタへのコールであるハイパーリンクが渡される。

【0048】図6は、状態情報をWebのハイパーリンクに保存するために動的引数組み込みを使用する方法を示す。ステップ601でクライアントはWebサイトのCGIスクリプト $p_1$ を起動する。

【0049】ステップ603でCGIスクリプト $p_1$ は、状態変数 $x_1, x_2, \dots, x_n$ をこれらの変数が、会話中に起動されるかもしれない全てのCGIスクリプトからアクセスできるように会話に組み込むことを決定する。CGIスクリプト $p_1$ は、クライアントが会話を続けるためにハイパーテキスト・リンクを持つHTMLページ $h$ を生成する。しかしページ $h$ をクライアントに

直接返す代わりに、 $p_1$ は、embed1と呼ばれる引数エンベッタのモジュールを起動する。CGIスクリプト $p_1$ はページ $h$ 及び状態変数 $x_1, x_2, \dots, x_n$ をembed1に受け渡す。

【0050】ステップ605でembed1モジュールは、ページ $h$ の全てのハイパーテキスト・リンクを変更し、embed2と呼ばれる引数エンベッタの他のモジュールを起動する。embed2に渡される引数は、ページ $h$ のリンクがHTMLファイルに対するものかCGIスクリプトに対するものかにより異なる。ページ $h$ のリンクがHTMLファイルに対するものなら、embed2には、ファイルへの絶対参照と全ての状態変数が渡される。ページ $h$ のリンクがCGIスクリプトに対するなら、embed2には、CGIスクリプトへの絶対参照、CGIスクリプトへのオリジナル引数、オリジナル引数の終わりを区切るパラメータ、及び状態変数が渡される。

【0051】例えばCGIスクリプト $p_1$ が、次の2つのハイパーリンクを含むHTMLページと共に、状態変数 $x1=32$ と $x2=45$ をembed1に受け渡すとする。

1) "http://www. watson. ibm. com/mail. html";及び  
2) "http://www. watson. ibm. com/cgi-bin/proq?arq1=55"

【0052】モジュールembed1はこれら2つのハイパーリンクをそれぞれ次のように変換する。

1) "http://www. watson. ibm. com/cgi-bin/embed2?url=/www. watson. ibm. com/mail. html&x1=32&x2=45";及び

2) "http://www. watson. ibm. com/cgi-bin/embed2?url=/www. watson. ibm. com/cgi-bin/proq&arq1=55&comma=1&x1=32&x2=45"

【0053】embed2は、2つ目のハイパーリンクの文字列"comma=1"により、オリジナル引数、arq1を状態変数、 $x1, x2$ から区別する。一意の任意の変数名をデリミッタとして使用できる。"comma"が同じ名前の変数と衝突するような危険がある場合は、より洗練された方法により、一意のデリミッタ変数名を拾うこともできよう。

【0054】ステップ607でembed1モジュールは、変更されたハイパーリンクで構成されたHTMLページ $h'$ をクライアントに送る。ページ $h'$ の全てのハイパーテキスト・リンクはembed2へのコールである。

【0055】ステップ609でクライアントはページ $h'$ からハイパーテキスト・リンクを選択して会話を続ける。ハイパーテキスト・リンクはモジュールembed2を起動する。

【0056】ステップ611でembed2はリクエストを処理する。embed2への引数がファイルを含む場合はファイルがフェッチされる。embed2への引数がCGIスクリプトへのコールを含む場合は、CGIスクリプトがオリジナル引数及び状態変数で起動される。上の例では、最初

のハイパーリンクがクライアントによって選択された場合、embed2はファイル"mail.html"をフェッチする。第2ハイパーリンクがクライアントによって選択された場合、embed2はオリジナル変数arg1=55と状態変数x1=32、x2=45でCGIスクリプト"prod"をコールする。

【0057】ステップ613で、対応するHTMLファイルがフェッチされるかCGIスクリプトが起動されると、得られる出力と状態変数はembed1に受け渡される。プロセスは次にステップ605に戻り、そこでembed1は出力のハイパーリンクに変更を加える。(動的引数組み込みの詳細はA. Iyengarによる1996年6月7日付、米国特許出願番号第08/660633号、"Preserving State in Stateless Network Protocols"を参照されたい。)

【0058】動的引数組み込みは、会話で状態を保存する好適な方法ではあるが、ネットスケープ・クッキー(NETSCAPE cookies)もまた使用できる。ネットスケープ・クッキーは非標準HTTP機能であり、HTTPオブジェクトをクライアントに返すサーバはこれにより、クライアントが格納する状態情報1つを送ることもできる。この状態情報に含まれるのは、状態情報が有効な範囲のURLの記述である。クライアントがネットスケープ・クッキーをサポートするなら、クライアントは、それが送るHTTPリクエストをチェックし、リクエストのURLがクライアントが格納している期限切れになっていないクッキーの範囲内かどうか確認する。HTTPリクエストがクッキーによって定義された範囲内なら、クライアントはそのクッキーに対応した状態情報をそのHTTPリクエストに加える。(ネットスケープ・クッキーの詳細については"Persistent Client State HTTP Cookies", Netscape Communications Corporation, 1996, [http://home.netscape.com/newsref/std/cookie\\_spec.html](http://home.netscape.com/newsref/std/cookie_spec.html)を参照されたい。)

【0059】ネットスケープ・クッキー上で動的引数組み込みを使用することにはいくつか利点がある。動的引数組み込みにより、全ての状態情報の存続期間が会話の存続期間に結び付けられる。クライアントとサーバ・グループの間で新しい会話が始まる毎に新しい状態情報が生成される。クッキーを使用すると、状態情報の存続期間を会話の存続期間に結び付けることは不可能である。クッキーの他の欠点は、標準HTTPプロトコルの一部ではないこと、よって全てのクライアント及びWebサーバと連携するとは限らないことである。対照的に、動的引数組み込みはHTTPをサポートする任意のクライアントまたはWebサーバと連携する。動的引数組み込みの他の利点は、クライアント側の異なるアカウントに対応したWebページをキャッシュすることによって複数のアカウントへのアクセスが同時に可能なことである。

【0060】HTML形式はまた状態を保存するのに

使用できる。HTML形式は、サーバが隠し変数を形式に組み込むために使用できる。隠し変数は、形式がクライアントによって提出されたときサーバに返される。しかしHTML形式の使用は制限され、厄介である。これはサーバが会話全体で状態情報を保存するために、動的に生成されたHTML形式でクライアントに常に応答しなければならないからである。

【0061】ステップ300でトランザクションが処理される。トランザクションが単純トランザクションなら、参加サーバまたはコントローラは自体でトランザクションを処理する。要求されたトランザクションがコントローラからのグループ・トランザクションなら、処理は図3のステップによって行われる。

【0062】ステップ209で単純トランザクションを処理するサーバは、会話を続けるためのハイパーテキスト・リンクと共にHTMLページをクライアントに返す。ページがクライアントに送られる前にページに状態情報を追加することもできる。例えばサーバ・グループが動的引数組み込みを使用している場合、ページ上のハイパーテキスト・リンクを先に述べたように変更しなければならない。サーバ・グループが動的引数組み込みを使用していない場合も、システムがクライアントの認証以降にアクセスしている参加サーバを管理できるように、ページのハイパーテキスト・リンクを変更しなければならないことがある。例えば、ページを送っていない参加サーバに対応したページのハイパーテキスト・リンクを変更して、アクセス・ノード状態変数403を必要に応じて更新するCGIスクリプトが起動される。

【0063】ステップ210でトランザクションが処理され、応答ページがクライアントに返されると、クライアントは、応答ページから新しいハイパーテキスト・リンクを選択することによって(場合によっては他の情報の入力と共に)会話を続ける。

【0064】ステップ211では、ステップ210で選択されたハイパーテキスト・リンクが参加サーバをポイントする場合、認証以降にサーバがアクセスしたかどうかCGIスクリプトによって確認される。これはアクセス・ノード状態変数403を調べることによって行われる。認証以降にサーバがアクセスしていない場合、サーバはステップ212でアクセス・ノード状態変数403に追加される。

【0065】ステップ213でクライアントからコンタクトされたサーバは、クライアントが認証を必要とするトランザクションを実行しようとしたか確認する。クライアントが実行しようとしなかった場合はトランザクションはステップ300で処理される。

【0066】認証を要するトランザクションが選択されていることをサーバが確認した場合は、処理はステップ214に続く。このステップでサーバは、ユーザID401及びセッションID402が有効か確認する。サー

バが参加サーバならコントローラとのコンタクトが、好適な実施例では、ユーザID及びセッションIDの有効性を確認するために必要である。ユーザID及びセッションIDが有効でない場合は、エラー処理がステップ215で行われる。先にステップ206に関して述べたとおり、エラー処理はいくつかの形で処理できるが、好適にはサーバがエラー・メッセージを含むページで応答することによって会話を終了させる。ユーザID及びセッションIDが有効ならトランザクションは先に述べたようにステップ300で処理される。

【0067】クライアントによって要求されたトランザクションがグループ・トランザクションなら、図3の本発明の好適な実施例に従って参加サーバ間の調整が行われる。

【0068】ステップ303でコントローラは、認証以降にアクセスした全ての参加サーバで構成される参加サーバのデフォルト・グループを構成する。この情報は、アクセス・ノード状態変数403に含まれる。クライアントは、デフォルト・グループを受け入れるか、または自体が選択する参加サーバのセットでデフォルト・グループを無効にするオプションを持つ。

【0069】グループ・トランザクションは、ある有限な $n > 0$ のとき $n$ 個のフェーズで起こり得る。最初の $n-1$ フェーズは準備フェーズと呼ばれる。最後のフェーズはコミット・フェーズと呼ばれる。ステップ304は準備フェーズをまだ実行すべきを確認する。準備フェーズを実行する必要がある場合、ステップ306でコントローラが対応する準備メッセージをステップ303で定義されたグループの全ての参加サーバに送る。

【0070】ステップ307でコントローラは、定義済みグループの全ての参加サーバから応答を集め、先のフェーズからのものにこれらの応答を加える。参加サーバは肯定応答または否定応答で応答する。参加サーバがある時間内に応答しなかった場合、応答は否定とみなされる。参加サーバは、それらの応答の一部としてHTMLテキストを送ることもできる。HTMLテキストの形式はフレキシブルなので、各参加サーバはグループ操作に対して、参加サーバが選択するハイパーテキスト・リンク、イメージ及びマルチメディア・コンテンツの組み合わせを含めて、カスタマイズされた応答を送ることができる。

【0071】ステップ309でコントローラは、全ての応答が肯定かどうかを確認する。参加サーバの応答が否定なら、処理はステップ311に移る。ステップ311でコントローラは、ステップ309で肯定応答した全ての参加サーバに、準備フェーズが失敗したことを通知する。

【0072】ステップ312でコントローラは、肯定応答した参加サーバからの応答を、失敗した準備フェーズ及び準備フェーズに失敗した参加サーバをクライアント

に通知するメッセージと共に1つのページにマージする。クライアントに戻るページはまた、クライアントが会話を続けるためのハイパーテキスト・リンクを含む。

【0073】コントローラがステップ309で、準備フェーズが定義済みグループの全ての参加サーバで成功したことを確認した場合は、処理はステップ304に戻る。準備フェーズが全て成功して完了した場合、処理はステップ305に進む。ステップ305でコントローラは、定義済みグループの全ての参加サーバにコミット・メッセージを送る。

【0074】ステップ308でコントローラは、定義済みグループの全ての参加サーバから応答を集める。先に述べたように、参加サーバはそれらの応答にHTMLテキストを追加できる。ステップ310でコントローラは1つのページの応答を作成する。これはいくつかのアイテムをマージすることによってクライアントに送られる。第1にコントローラは、先にステップ307で準備フェーズから集められた応答とステップ308で受信された応答を全てマージする。コントローラは次に、全ての準備フェーズが成功したことを伝えるメッセージを追加する。参加サーバがステップ308で時間内に応答することができなかった場合、コントローラはまたこの情報を応答ページに追加することができる。最後にクライアントが会話を続けるられるようにするハイパーテキスト・リンクが追加される。ここでもページがクライアントに送られる前に状態情報を保存するために、またはシステムが先にアドレスされたノードを管理できるように、ページの変更が必要になることがある。

【0075】上に述べた実施例ではいくつか拡張及び変形が可能である。1つの変形例はセッションIDの存続期間を制限することである。コントローラは各セッションIDの存続期間及び作成時刻（エフェメラル・セッションIDと呼ばれる）を、ステップ208でセッションIDが作成されたとき記録できる。現在時刻及び作成時刻の違いが、エフェメラル・セッションIDの存続期間を超えるときは、エフェメラル・セッションIDは期限を過ぎていることになる。エフェメラル・セッションIDの期限切れはステップ214でチェックできる。クライアントが期限の切れたエフェメラル・セッションIDでトランザクションを試行する場合は、ステップ215でエラー処理が行われる。期限切れエフェメラル・セッションIDを処理する好適な方法は、有効なユーザID及びパスワードを提出することで自体を再認証することをクライアントに要求することである。

【0076】他の変形例として、各トランザクション・リクエストに新しいセッションIDを作成することが考えられる。この場合、各セッションIDは1つのトランザクションにのみ有効である。このアプローチでは、悪意ある侵入者がセッションIDを盗んでアカウントに侵入することは難しくなる。この変形例を実現するために

10

20

30

40

50

は、ステップ214を変更して前のセッションIDを無効にし、新しいセッションIDを作成して新しいセッションIDを記録し、新しいセッションIDを会話で保存することができる。

【0077】他の変形例は、セッションIDをクライアントに送られる前に暗号化し、コントローラによって認証される前にデコードすることである。動的引数組み込み、クッキー等の手法が用いられる場合は、この暗号化機能によりセッションIDの受け渡し先であるURLにもとづく異なるエンコーディングを作ることができる。従って異なるURLへのアクセスには異なる暗号化されたセッションIDが必要になる。暗号化されたセッションIDを何とか盗もうとした侵入者は、それを特定のURL（また場合によっては特定の引数セットしかないURL）でしか使用できないことになる。これによりこのような侵入者による損害が制限される。

【0078】他の変形例は、コントローラが特定のセッションIDからの全てのトランザクションの出所をセッションIDを作成した認証プロセスに参加したクライアントと同じIPアドレスに限定することである。この機能を実現するためには、セッションIDがステップ208で作成されるときに自体を認証するクライアントのIPアドレスが記録される。ステップ214で、リクエストを出すクライアントのアドレスが、自体を認証したクライアントのアドレスと同じ場合にのみトランザクションを行うことができる。このアプローチでは、悪意あるユーザがクライアントからセッションIDを盗み、セッションIDを使用してクライアントのアカウントに他のマシンから侵入することを防ぐことはできる。しかし絶対確実ではない。異なるマシンが同じIPアドレスを持つように見えることがあるからである（例えばマシンが同じファイアウォール下にあるとき）。暗号化等他のセキュリティ機能を、このアプローチと併用してアカウント侵入の危険を更に小さくすることは可能である。

【0079】他の変形例は、コントローラによって維持されるクライアントの認証及びアカウント情報を1つ以上の参加サーバに複製することである。その場合、コントローラは認証毎にコンタクトする必要はない。このアプローチでは、コントローラの負荷が減り、システム性能が向上する。しかし認証及びアカウントの情報の複数のコピーの間で一貫性を保つためには何らかの方法が必要になる。

【0080】他の変形例は、作業負荷が大きいときにシステム性能を改良するもので、複数のサーバで実行するためにコントローラをスケーリングすることである。しかし複数のサーバで実行するためにコントローラをスケーリングする際の複雑さは、1つは、複数のノードで認証情報をどのように共有するかという課題である。認証情報は複数のノードに複製できるが、複数の一貫した認証情報のコピーを維持するオーバーヘッドと複雑さのため

に、ステップ208で記録された認証情報のコピーは1つだけ維持することが望ましい場合もある。

【0081】コントローラが複数のノードでスケーリングされるときに認証情報を維持する1つの方法は、（サーバ・グループと現在通信している全てのクライアントに属する）全てのセッションIDのセットをコントローラ・ノードに分散させることである。従って、あるセッションIDのコピー1つだけが維持されるが、セッションIDのセットはコントローラ・ノードに分散される。このアプローチを実現するために、コントローラは、あるクライアントのセッションID情報をステップ208でコントローラ・ノードnに格納できる。コントローラ・ノードnのIDは次に、そのクライアントとの会話で状態変数にエンコードできる。例えばセッションID変数402をエンコードして、実際のセッションID及びクライアントのセッションIDを格納したコントローラのノードの両方を格納できよう。クライアントが、そのセッションIDの認証を要するトランザクションを実行しようとしたときは、ステップ214で、クライアントとの会話で保存された対応する状態変数が調べられ、クライアントのセッションIDがどのコントローラ・ノードにあるか確認される。クライアントのセッションIDは次に対応するコントローラ・ノードで確認される。

【0082】クライアント認証情報と同じく、クライアント・アカウント情報も複数のコントローラ・ノードに複製するか、または複数のコントローラ・ノードに分散させることができる。どのアプローチを取るかの決定は、クライアント認証情報についてどちらのアプローチを使用するかとは別に行える。クライアント・アカウント情報を複数のノードに分散させ場合、上に述べたアプローチが使用できる。あるクライアントについてアカウント情報を含むコントローラ・ノードのIDは、ユーザID変数401等の状態変数にエンコードできる。

【0083】他の変形例は、通常はサーバ・グループによって実行される機能のいくつかをクライアントが実行できるようにすることである。Java等の言語を使用して、サーバ・グループによって実行される機能のいくつかをダウンロードしてクライアントによって実行できる。

【0084】他の変形例は、会話でユーザIDとセッションID両方の状態変数を保存する代わりに、セッションID状態変数だけを保存することである。この例でサーバは、ユーザIDとセッションIDの対応を維持することになる。次にサーバはセッションIDからユーザIDを計算することができる。

【0085】他の変形例は、セッションID状態変数を作成する代わりにクライアントのパスワードを状態変数として維持することである。しかしセッションIDは会話が続く間しか続かないので、パスワードの代わりにセッションIDを使用する利点は、盗まれたセッションID

Dは盗まれたパスワードよりも潜在的に害が少ない場合があるということである。

【0086】他の変形例は、多数フェーズ・コミットを扱うためにXAプロトコルを実現することである。(XAプロトコルの詳細についてはJ. Gray & A. ReuterによるTransaction Processing: Concepts & Techniques, Morgan Kaufmann, 1993を参照されたい)。

【0087】好適な実施例では、グループ・トランザクションの参加サーバのデフォルト・グループが、認証時刻とグループ・トランザクションが要求された時刻の間10にアクセスした参加サーバで構成される。デフォルト・グループの選択についてはいくつかの変形例を簡単に実現できる。参加サーバのデフォルト・グループは次のサーバで構成することができる。1) 会話の初め以降にアクセスした全てのサーバ、2) 会話の初め以降に単純トランザクションに関係した全てのサーバ、3) 最後のグループ・トランザクション以降にアクセスした全てのサーバ、4) 最後のグループ・トランザクション以降に単純トランザクションに関係した全てのサーバ、及び5) 最後のグループ・トランザクションに参加した全てのサーバ。またこれらを組み合わせたデフォルト・グループも可能である。

【0088】一般的にデフォルト・グループは、トランザクションのタイプ、認証のタイプ及び重要イベントに関して決定することができる。トランザクションは、1) 全てのトランザクション、2) 単純トランザクション、及び3) グループ・トランザクションを含めたグループに分類することができる。トランザクションは更に、1) 全てのトランザクション、2) 認証が必要なトランザクション、及び3) 認証が必要でないトランザク30ション等の認証タイプに従って分類できる。重要イベントは、トランザクションによって定義された会話中の1時点である。例えば次は重要イベントになる場合がある。1) 会話の初め、2) クライアントが最初に有効な識別情報を与える時点、3) 最後のグループ・トランザクション。デフォルト・グループは、サーバの重要なサブセット、CS (i, j, k) に関して定義することもできる。これはトランザクション・タイプjと認証タイプkのトランザクションを与えた重要イベントi以降にクライアントが通信したサーバのサブセットである。例40例えば上に定義した分類によると、サーバの重要サブセットCS (3, 1, 2) は、認証を要するトランザクションが実行された最後のグループ・トランザクション以降にクライアントが通信したサーバのセットである。

【0089】他の変形例は、アクセスした参加サーバに依存しないデフォルト・グループを指定することである。この例ではアクセス・ノード状態変数403を保存する必要はなく、ステップ211、212も必要ない。

【0090】参加サーバが複数のコントローラと通信することも可能である。図4に示すように、1つのコント

ローラが、階層の下レベルのコントローラと参加サーバからの操作を調整する、コントローラの階層を持たせることも可能である。

【0091】図3のステップ310、312の好適な実施例では、コントローラが全ての情報を1つのHTMLページとしてクライアントに送る。情報の量が多い場合は、これに代わる方法として、情報をいくつかのページに分けることが考えられる。コントローラは一度に1ページを送る。各ページにハイパーテキスト・リンクがあり、クライアントはこれにより他のページを見ることができる。ハイパーテキスト・リンクを(可能な限りいくつかを)辿ることにより、クライアントは全ての情報を見ることができる。この技術はもちろん新しいものではない。例えばURL、<http://www.lycos.com>のLycosサーチ・エンジンはこの方法を実現している。

【0092】他の変形例は、手動により識別情報を入力することをクライアントに促す代わりに、クライアントを識別するためにデジタル認証を用いることである。デジタル認証は、クライアントのディスクまたはスマート・カードに格納できる識別情報の文字列である。デジタル認証は、クライアントに手動での入力を期待できるものより長く複雑になる情報の文字列で構成できるので、デジタル認証を使用することはクライアントに識別情報を手動で入力させる場合よりも、クライアントを識別する方法としてより安全な方法を提供できる場合がある。またデジタル認証及び手動入力を組み合わせて使用することもできる。例えばクライアントには、そのスマート・カードを有効にするためにPIN (個人識別番号) の入力を要求することができる。

【0093】上に述べた好適な実施例で、クライアント・アカウント情報はコントローラ・サーバによって維持される。他の変形例は、コントローラが、クライアントによって与えられたクライアント識別情報を確認するために認証機関と通信することである。

【0094】まとめとして、本発明の構成に関して以下の事項を開示する。

【0095】(1) 1つ以上の参加サーバ及び1つ以上のコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント/サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡す手順が制限されたまたは未定義のプロトコルを使用して前記クライアント及び前記サーバの間で通信するステップと、前記クライアントから前記サーバの1つに操作のリクエストを転送するステップと、前記リクエストに応答して複数のサーバ及び少なくとも1つのコントローラ・サーバの間で通信して少なくともいくつかのサーバの操作を調整するステップと、を含む、方法。

(2) 参加サーバとコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント／サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡す手順が制限されたまたは未定義のプロトコルを使用して前記クライアントと前記サーバとの間で通信するステップと、認証を要するクライアントによる第1操作のリクエストを転送するステップと、前記クライアントにより

10 クライアント識別情報を与えるステップと、前記クライアント識別情報をコントローラ・サーバにより確認するステップと、コントローラ・サーバによりクライアント認証情報を作成するステップと、サーバにより前記クライアント認証情報を維持するステップと、前記クライアント認証情報を会話で保存するためにサーバにより手段を利用するステップと、認証を要する前記サーバによる第2操作のリクエストを前記サーバの1つに転送するステップと、前記会話で保存された前記クライアント認証情報とサーバによって維持されたクライアント認証情報

20 を使用して前記クライアントを認証するステップと、を含む、方法。

(3) 1つ以上の参加サーバ及び1つ以上のコントローラ・サーバを含み、各参加サーバが少なくとも1つのコントローラ・サーバと通信できるサーバ・グループを持つクライアント／サーバ環境にて、前記サーバ・グループのサーバ間で操作を調整する方法であって、クライアント及び前記サーバ・グループの少なくとも1つのサーバの間で情報を受け渡す手順が制限されたまたは未定義のプロトコルを使用して前記クライアント及び前記サーバの間で通信するステップと、参加を要する操作のリクエストを複数のサーバにより前記クライアントから前記サーバの1つに転送するステップと、前記リクエストに

30 応答して前記サーバと少なくとも1つのコントローラ・サーバの間で通信して前記複数のサーバの操作を調整するステップと、を含む、方法。

(4) クライアントによりクライアント識別情報を与える前記ステップは前記クライアントによる情報の手動入力により行われる、前記(2)記載のサーバ間で操作を調整する方法。

(5) クライアントによりクライアント識別情報を与える前記ステップは永続メモリに格納された情報を取得することにより行われる、前記(2)記載のサーバ間で操作を調整する方法。

(6) 永続メモリに格納された情報はディスクまたはスマート・カード上の1つ以上のデジタル認証を含む、サーバ間で操作を調整する、前記(5)記載の方法。

(7) クライアント識別情報を確認する前記ステップは、1つ以上のコントローラ・サーバによって維持されたクライアント・アカウント情報を取得し、前記クライ

アント識別情報を前記クライアント・アカウント情報と比較するステップを含む、サーバ間で操作を調整する、前記(2)記載の方法。

(8) クライアント識別情報を確認する前記ステップは、前記サーバ・グループの外部の認証機関サーバと通信するステップを含む、サーバ間で操作を調整する、前記(2)記載の方法。

(9) 1つ以上のサーバにより前記クライアント認証情報を維持するステップは、前記クライアント認証情報を前記サーバ・グループの2つ以上のサーバに複製するステップを含む、サーバ間で操作を調整する、前記(2)記載の方法。

(10) 前記クライアント認証情報を会話で保存する手段は動的引数組み込みである、サーバ間で操作を調整する、前記(2)記載の方法。

(11) 前記クライアントはWorld Wide Webクライアントであり、前記サーバはWorld Wide Webサイトであり、前記クライアント認証情報を会話で保存する手段はネットスケープ・クッキー(NETSCAPE cookie)である、サーバ間で操作を調整する、前記(2)記載の方法。

(12) 前記クライアントはWorld Wide Webクライアントであり、前記サーバはWorld Wide Webサイトであり、前記クライアント認証情報を会話で保存する手段はHTML形式である、サーバ間で操作を調整する、前記(2)記載の方法。

(13) クライアント認証情報を維持する前記サーバは少なくとも1つのコントローラ・サーバを含む、サーバ間で操作を調整する、前記(2)記載の方法。

(14) 前記クライアント認証情報はセッションIDを含む、サーバ間で操作を調整する、前記(2)記載の方法。

(15) 前記セッションIDは、経過すると有効ではなくなる存続期間を持つ、サーバ間で操作を調整する、前記(14)記載の方法。

(16) 前記クライアントにより、認証を要する指定数のリクエストを1つ以上のサーバに転送するステップと、前記指定数のリクエストの間に1つ以上のコントローラ・サーバにより新しいセッションIDを含む新しいクライアント認証情報を作成するステップと、1つ以上のサーバにより前記新しいクライアント認証情報を会話で保存する手段を利用するステップと、を含む、サーバ間で操作を調整する、前記(14)記載の方法。

(17) クライアント認証情報を保存する手段は、前記セッションIDを暗号化する手段を含み、前記会話で保存されたクライアント認証情報を使用することは暗号化されたセッションIDのデコードを伴う、サーバ間で操作を調整する、前記(14)記載の方法。

(18) 前記セッションIDを暗号化する手段は、サーバとリクエストの特定の組み合わせについてののみ有効な暗号化されたセッションIDを作成する、サーバ間で操

作を調整する、前記(17)記載の方法。

(19) 前記サーバ・グループのサーバはWorld Wide Webサイトであり、前記暗号化されたセッションIDは特定のURLについてのみ有効である、サーバ間で操作を調整する、前記(18)記載の方法。

(20) 少なくとも1つのサーバにより前記クライアント識別情報を与えた前記クライアントのアドレスを維持するステップと、前記会話中に認証を要し前記クライアントにより要求された少なくともいくつかの操作にตอบสนองして、前記アドレスを使用して前記クライアントを認証するステップと、を含む、サーバ間で操作を調整する、前記(2)記載の方法。

(21) 少なくとも1つのコントローラ・サーバは複数のノードに分散される、サーバ間で操作を調整する、前記(2)記載の方法。

(22) 前記クライアント認証情報はセッションIDを含み、1つ以上サーバにより前記クライアント認証情報を維持するステップは、前記セッションIDを前記分散されたコントローラ・サーバの1つの全てのノードよりも少ないノードに格納し、前記セッションIDが格納された前記ノードの少なくとも1つのIDを前記会話で状態変数に保存するステップを含む、サーバ間で操作を調整する、前記(21)記載の方法。

(23) 前記クライアント認証情報は前記分散されたコントローラ・サーバの1つの複数のノードに複製される、サーバ間で操作を調整する前記(21)記載の方法。

(24) 少なくとも1つのコントローラ・サーバが複数のノードに分散され、前記クライアント・アカウント情報は前記分散されたコントローラ・サーバの1つの複数のノードに複製される、サーバ間で操作を調整する、前記(7)記載の方法。

(25) 前記複数のサーバの操作を調整するステップは、前記クライアントによるリクエストにตอบสนองして少なくとも1つのコントローラ・サーバによりサーバのデフォルト・グループを定義するステップを含む、サーバ間で操作を調整する、前記(3)記載の方法。

(26) 前記サーバのデフォルト・グループは、前記クライアントが会話中に通信した前記サーバ・グループのサーバを含む、サーバ間で操作を調整する、前記(25)記載の方法。

(27) 前記サーバのデフォルト・グループは、サーバの重要サブセットに関して決定され、サーバの各重要サブセットは、トランザクション・タイプ、認証タイプ及び重要イベントに関して決定される、サーバ間で操作を調整する、前記(26)記載の方法。

(28) 前記トランザクション・タイプは全てのトランザクション、単純トランザクション、及びグループ・トランザクションを含む、サーバ間で操作を調整する、前記(27)記載の方法。

(29) 前記認証タイプは、全てのトランザクション、認証を要するトランザクション及び認証を要しないトランザクションを含む、サーバ間で操作を調整する、前記(27)記載の方法。

(30) 前記重要イベントは、会話の初め、クライアントが最初に有効な識別情報を与える時点、及び最後のグループ・トランザクションを含む、サーバ間で操作を調整する、前記(27)記載の方法。

(31) 前記サーバのデフォルト・グループは、前記最後のグループ・トランザクションに関係したサーバに関して決定される、サーバ間で操作を調整する前記(25)記載の方法。

(32) 前記操作のリクエストは複数のサーバから情報を集める操作を伴う、サーバ間で操作を調整する、前記(3)記載の方法。

(33) 複数のサーバの参加を要する前記操作のリクエストは、有限な $n > 0$ のとき複数のサーバからの $n$ フェーズ・コミットを伴う、サーバ間で操作を調整する、前記(3)記載の方法。

(34) 前記サーバ間で少なくとも1つのコントローラ・サーバと通信して前記複数のサーバの操作を調整するステップは、少なくとも1つのコントローラ・サーバ及び前記複数のサーバのうち少なくともいくつかのサーバとの間で通信するステップと、前記複数のサーバのうち少なくともいくつかのサーバからの応答を少なくとも1つのコントローラ・サーバによりマージするステップと、を含む、サーバ間で操作を調整する、前記(3)記載の方法。

(35) 前記サーバはWorld Wide Webサイトである、サーバ間で操作を調整する、前記(1)乃至前記(3)のいずれかに記載の方法。

(36) 前記クライアント及びサーバの間の通信に用いられる前記プロトコルはハイパーテキスト転送プロトコルである、サーバ間で操作を調整する、前記(1)乃至前記(3)のいずれかに記載の方法。

(37) 前記クライアント及びサーバの間の通信に用いられる前記プロトコルは安全である、サーバ間で操作を調整する、前記(1)乃至前記(3)のいずれかに記載の方法。

(38) 少なくとも1つの参加サーバは複数のコントローラと通信する、サーバ間で操作を調整する、前記(1)乃至前記(3)のいずれかに記載の方法。

(39) コントローラ・サーバの階層が存在し、各コントローラ・サーバが下の階層レベルの参加サーバ及びコントローラ・サーバの間で操作を調整する、サーバ間で操作を調整する、前記(1)乃至前記(3)のいずれかに記載の方法。

(40) 前記クライアントにより少なくとも1つのサーバからプログラムを受信し実行することによって、前記クライアントが前記サーバ・グループの機能の少なくとも

もいくつかを実行するステップを含む、サーバ間で操作を調整する前記(1)乃至前記(3)のいずれかに記載の方法。

(41) 前記参加サーバは互いに通信できない、サーバ間で操作を調整する前記(1)乃至前記(3)のいずれかに記載の方法。

【図面の簡単な説明】

【図1】本発明の好適な実施例に従って用いられるアーキテクチャ全体を示す図である。

【図2】本発明の好適な実施例に従ってクライアントの 10 認証を調整する方法を示す図である。

【図3】本発明の好適な実施例に従って参加サーバ間でグループ・トランザクションを調整する方法を示す図である。

\*

【図4】本発明の好適な実施例に従って用いられる他のアーキテクチャ全体を示す図である。

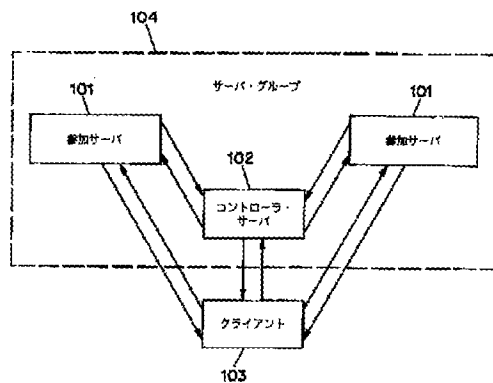
【図5】会話で保存される好適な変数のリストを示す図である。

【図6】状態情報をWWWハイパーリンクに動的に組み込む方法を示す図である。

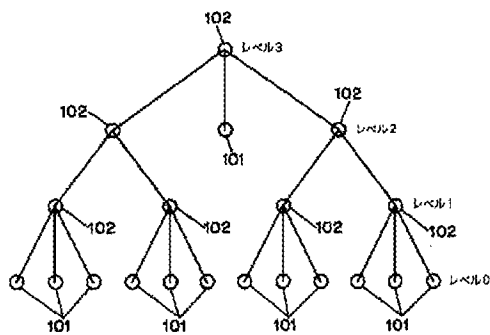
【符号の説明】

101 参加サーバ  
102 コントロール・サーバ  
103 クライアント  
104 サーバ・グループ  
401 ユーザID  
402 セッションID  
\* 403 アクセス・ノード状態変数

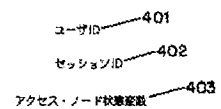
【図1】



【図4】

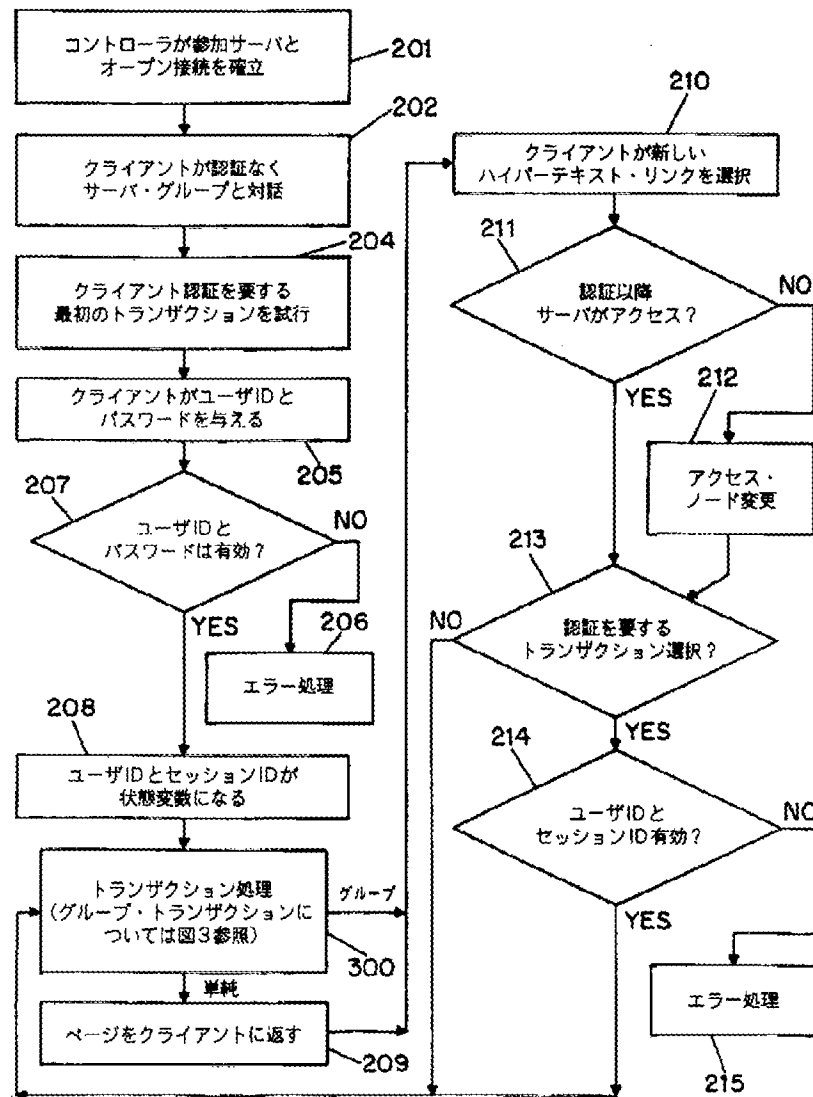


【図5】

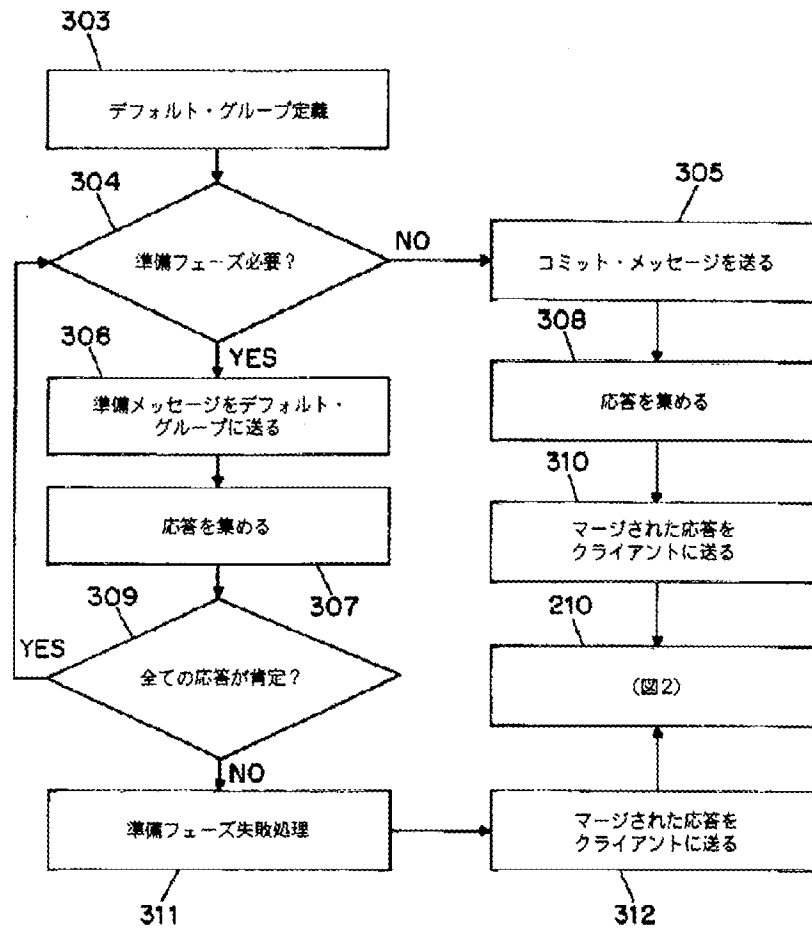




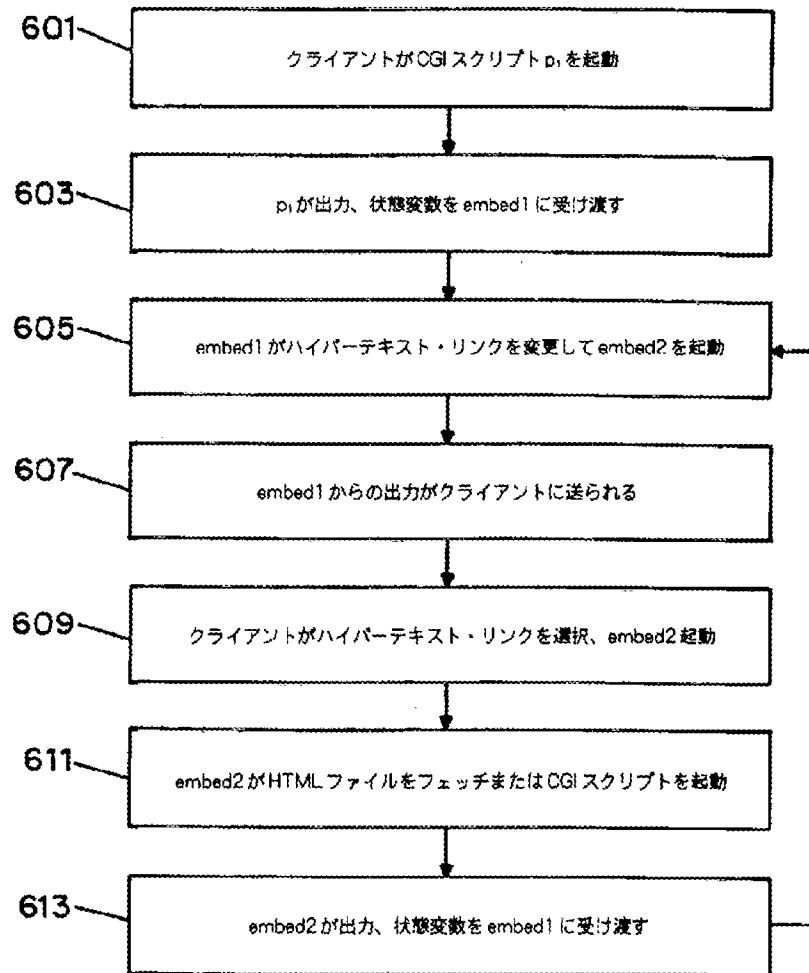
【図2】



【図3】



【図6】



フロントページの続き

(72)発明者 アラン・ワンギル・イヤンガー  
アメリカ合衆国10598、ニューヨーク州ヨ  
ークタウン・ハイツ、パーク・レーン  
1160